ON THE COMPLEXITY OF COMPUTING GRAPH ISOMORPHISM

by

D. Hirschberg
M. Edelberg

Technical Report TR-130

August 1973

Computer Science Laboratory
Department of Electrical Engineering
Princeton University
Princeton, New Jersey 08540

Research supported by NSF Grant GJ-30126 and a National Science Foundation Graduate Fellowship.

ON THE COMPLEXITY OF COMPUTING GRAPH ISOMORPHISM

D. Hirschberg
M. Edelberg

INTRODUCTION

The notion of (polynomial) complete problems was introduced by Cook [1] and Karp [4]. A problem P is polynomial if there exists a Turing machine T and a polynomial Q(x) of finite degree such that, for all instances P_i of P, P_i can be described in space n implies that P_i can be solved by T within a time-bound of Q(n). It is known that if any one of the set of complete problems is polynomial then all of them are. Conversely, if it can be shown that any one of the set of complete problems is not polynomial, then none of them are.

There are several problems that are "no worse" than complete, but have not been shown to be either polynomial or complete. Graph isomorphism (given graphs A and B, is A isomorphic to B?) is one such.†

We discuss some approaches that attempt to categorize this problem. We also describe and refute various conjectures and algorithms for this problem.

† Another is k-processor scheduling of equal execution time, non-preemptible jobs with precedence constraints.

CANONICAL FORM OF A GRAPH

Consider a mapping $f: \mathcal{A} \to \mathcal{C}$ from the set $\mathcal{A}$ of adjacency matrices (representing labeled undirected graphs on the set of nodes $\{1,2,\ldots,n\}$) to an arbitrary set $\mathcal{C}$. Let $A_1 \in \mathcal{A}$ represent graph G_1 and $A_2 \in \mathcal{A}$ represent graph G_2. If G_1 is isomorphic to G_2 (written $G_1 \cong G_2$) then we shall say that A_1 is equivalent to A_2 (written $A_1 \equiv A_2$). Formally, $A_1 \equiv A_2$ IFF there exists a permutation matrix π such that $A_2 = \pi^{-1} A_1 \pi$.

We define

f is <u>isomorphism invariant</u> $A_1 \equiv A_2 \Rightarrow f(A_1) = f(A_2)$

f is <u>isomorphism indicative</u> $f(A_1) = f(A_2) \Rightarrow A_1 \equiv A_2$

f is an <u>isomorphism characterization</u>

or a <u>canonical mapping</u> $f(A_1) = f(A_2) \Leftrightarrow A_1 \equiv A_2$

in which case $f(A_1)$ is the <u>canonical form</u> of the graph represented by A_1.

Any algorithm which computes a canonical mapping for graphs can be used as a graph isomorphism algorithm.

In considering an adjacency matrix A, we need only look at the upper triangular part (u.t.p.) since A is a symmetric 0-1 matrix with zeros on the diagonal (we are considering simple undirected graphs with no self-loops).

We define an order on the bit positions of the u.t.p. of A as a mapping $R: \{(i,j) \mid 1 \le i < j \le n\} \Rightarrow \{1,2,\ldots,N=\frac{n^2-n}{2}\}$. An example of a possible R is given in figure 1.

We define a measure $g_R(A)$ by

$$g_R(A) = \sum_{1 \le i < j \le n} a_{ij} \cdot 2^{N-R(i,j)}$$

where a_{ij} is an element (0 or 1) in adjacency matrix A and $R(i,j)$ is the rank of a_{ij} according to R. It is seen that $g_R(A)$ is the base 2 evaluation of the bits in the u.t.p. of A arranged in order R. For fixed n, $g: \mathcal{A} \rightarrow \{0,1,2,\ldots,2^N-1\}$ is bijective. Let

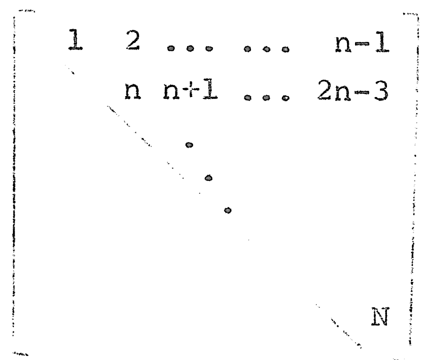


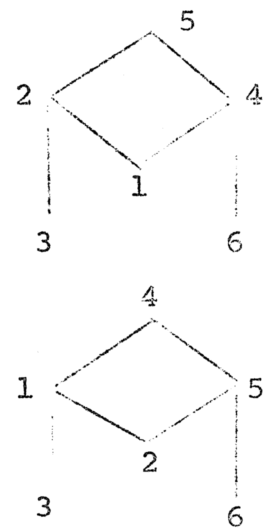
FIGURE 1

$$\#_R(A) = \max_{A_i \equiv A} (g_R(A_i))$$

Then $\#_R$ is a canonical mapping for graphs. For suppose $\#_R(A_1) = k = \#_R(A_2)$. Then $A_1 \equiv g^{-1}(k) \equiv A_2$ and thus $A_1 \equiv A_2$. Conversely, $A_1 \equiv A_2$ implies that $\max_{A_i \equiv A_1} (g_R(A_i)) = \max_{A_i \equiv A_2} (g_R(A_i))$ because $\equiv$ is an equivalence relation.

A neighborhood search algorithm is an algorithm that seeks to maximize a measure by, for a given starting point, looking at points that are "close" (in some sense) and choosing some point in the neighborhood of points which has a larger measure. This procedure is iterated until a local optimum is reached. In some cases (convex programming is one) a local optimum is always a global optimum. In other cases (as below) it may not be a global optimum.

In particular, for R as in figure,1, using simple row (and column) interchange will not necessarily yield a global optimum of $g_R(A)$. The two graphs in figure 2 are isomorphic and their matrix representations are stable under interchange (any single row interchange with corresponding column interchange will not yield a higher value of $g_R(A)$, yet they have unequal measures).




```
11100
 1010
  000
   11
    0
```

```
11100
 1010
  000
   10
    1
```

FIGURE 2

Theorem There exists an R such that determining whether $\#_R(G_1) \geq \#_R(G_2)$ is a polynomial complete problem.

Proof Take R as in figure 3 and take G_2 to be a circuit on n points. Thus $\#_R(G_2) = (\underbrace{11...1}_{n}00...0)_2$.

If G_1 has a Hamiltonian circuit then we can order the nodes of G_1 to that those bit positions ranked 1 through n by R have 1's and so $\#_R(G_1) \geq \#_R(G_2)$. If $\#_R(G_1) \geq \#_R(G_2)$ then those n bit positions must all have 1's and therefore there is a Hamiltonian circuit in G_1. Thus the Hamiltonian circuit problem reduces to determining whether

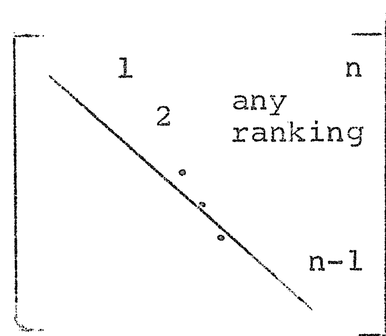


FIGURE 3

$\#_R(G_1) \geq \#_R(G_2)$ for this R (which is in NP) and we have shown the polynomial completeness of this problem. ∎

SPECIAL CASES

There are polynomial algorithms for various restrictions of the graph isomorphism problem. Tree isomorphism has a linear algorithm [3,7], planar graphs $O(n \log n)$ [3], graphs having no strongly regular subgraph in $O(n^5)$ [2].

We look at another "special case," bipartite graphs, and show that it is polynomial if and only if general graph isomorphism is polynomial.

<u>Theorem</u> General graph isomorphism red† bipartite graph isomorphism.

<u>Proof</u> Given $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ where G_i is a graph on vertex set V_i and edge set E_i, we shall construct $\hat{G}_1 = (\hat{V}_1, \hat{E}_1)$ and $\hat{G}_2 = (\hat{V}_2, \hat{E}_2)$ such that

1. $\hat{G}_1$, $\hat{G}_2$ are bipartite
2. $|\hat{V}_i| \leq |V_i|^2$
3. $(G_1 \cong G_2) \Leftrightarrow (\hat{G}_1 \cong \hat{G}_2)$

Let

$$\hat{V}_i = V_i \cup E_i \quad (i=1,2)$$
$$\hat{E}_i = \{\{j,\{j,k\}\} \mid j \in V_i \wedge \{j,k\} \in E_i\}$$

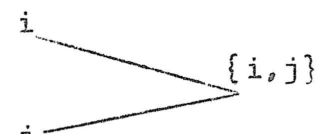


FIGURE 4

†Denotes the relation "is reducible to," see Karp [4].

1. $\hat{G}_i$ is bipartite (V_i, E_i are the two independent blocks)

2. $|\hat{V}_i| = |V_i| + |E_i| \le |V_i|^2$

3. (a) $(G_1 \cong G_2) \Rightarrow (\hat{G}_1 \cong \hat{G}_2)$

   assume $\exists f: V_1 \to V_2$ s.t. $\{i,j\} \in E_1 \Leftrightarrow \{f(i), f(j)\} \in E_2$

   Then define $\hat{f}: \hat{V}_1 \to \hat{V}_2$ as follows:

   $\hat{f}(j \in V_1) = f(j)$

   $\hat{f}(\{j,k\} \in E_1) = \{f(j), f(k)\}$

   Clearly $\hat{f}$ is an isomorphism.

   (b) $(\hat{G}_1 \cong \hat{G}_2) \Rightarrow (G_1 \cong G_2)$

   assume $\exists\ \hat{f}: \hat{V}_1 \to \hat{V}_2$ which is an isomorphism.

   <u>CASE 1</u> $\hat{G}_1$ is connected

   $\hat{G}_1$ is bipartite and there are only two ways to properly 2-color $\hat{V}_1$. Assume $|V_1| \ne |E_1|$, then $\hat{f}(V_1) = V_2$, $\hat{f}(E_1) = E_2$.

   Define $f: V_1 \to V_2$ by $f(v) = \hat{f}(v)$.

   f is an isomorphism from G_1 to G_2, since

   $$\begin{aligned}\{i,j\} \in E_1 &\Leftrightarrow \{i,\{i,j\}\}, \{j,\{i,j\}\} \in \hat{E}_1 \\ &\Leftrightarrow \{\hat{f}(i), \hat{f}(\{i,j\})\}, \{\hat{f}(j), \hat{f}(\{i,j\})\} \in \hat{E}_2 \\ &\Leftrightarrow \{f(i), f(j)\} \in E_2\end{aligned}$$

   If $|V_1| = |E_1|$ then G_1 is a graph with a single circuit. If G_1 is not a circuit, then $\hat{f}(V_1) = V_2$ because there exists a $v \in V_1$ with degree ≥ 3 forcing $\hat{f}(v) \in V_2$ (see figure 5). If the graph is a circuit then there is an isomorphism $\hat{f}$ which will map V_1 to V_2

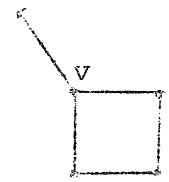


FIGURE 5

CASE 2 $\hat{G}_1$ is not connected

Let $\hat{G}_1 = \hat{H}_{11} + \hat{H}_{21} + \ldots + \hat{H}_{m1}$ where $\hat{H}_{i1}$ is a connected graph $(1 \le i \le m)$

Then $\hat{f}(\hat{H}_{i1}) = \hat{H}_{i2}$ (a connected graph) and $\hat{H}_{i2}$ must be bipartite. From case 1 we conclude that $(\hat{H}_{i1} \cong \hat{H}_{i2}) \Rightarrow (H_{i1} \cong H_{i2})$ therefore $(\hat{G}_1 \cong \hat{G}_2) \Rightarrow (G_1 \cong G_2)$.

SOME ISOMORPHISM CHARACTERIZATION APPROACHES

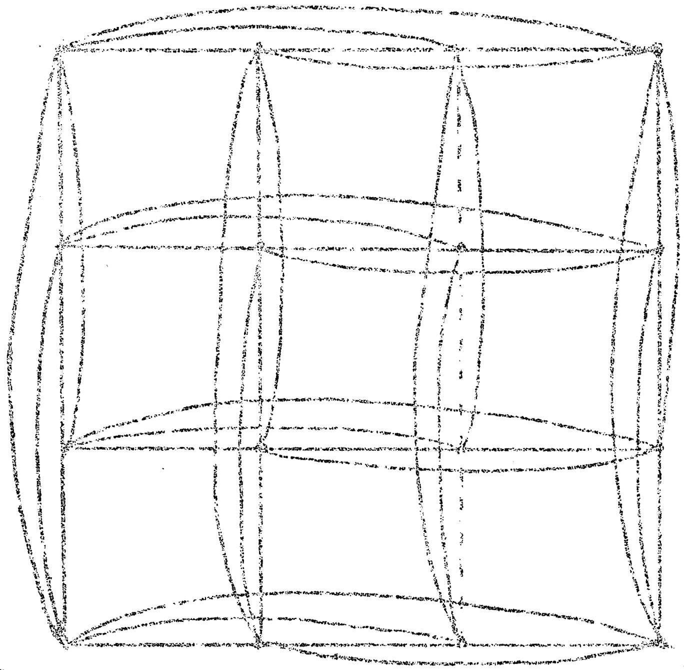
Walk Statistics

A k-walk from node i to node j is a sequence of vertices $(v_0, v_1, \ldots, v_k)$ such that $v_0 = i$, $v_k = j$ and v_ℓ adjacent to $v_{\ell+1}$ for all $0 \le \ell < k$.

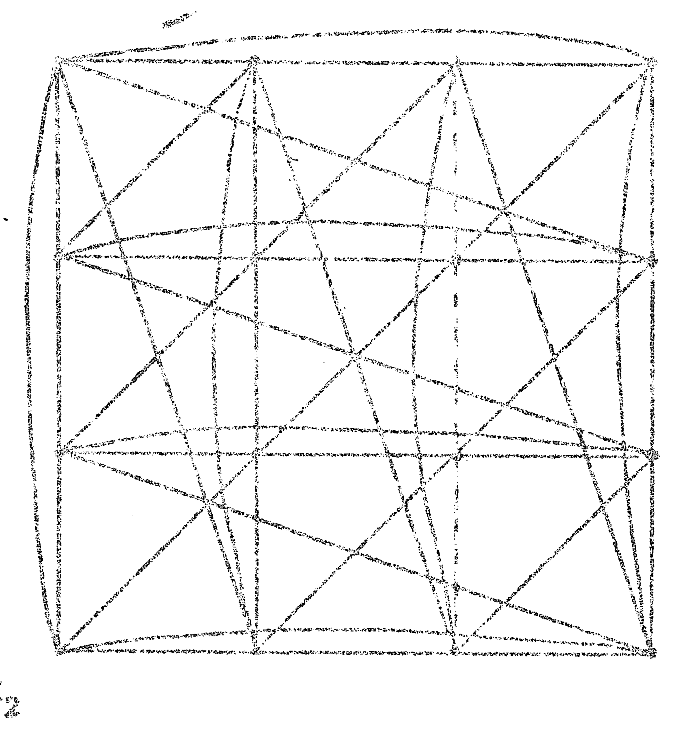
Define B_k as the matrix of k-walk counts. That is, $B_k(i,j)$ will be the number of distinct k-walks from node i to node j. Then, $B_0 = I$ (identity matrix), $B_1 = A$ (adjacency matrix), and $B_k = A^k$ (over the ring of integers). Since $B_{k+1}(i,j) = \sum_{\ell \text{ adj } j} B_k(i,\ell)$ we have $B_{k+1} = AB_k$.

Let $D = (B_0, B_1, \ldots, B_m)$ where m is an arbitrary number. Perhaps we can use the multi-set of numbers appearing in D as a canonical form. To yield a polynomial procedure, m can even be n^{100}.

In order to show that walk statistics are not sufficient to characterize graphs, we introduce graphs H_1 and H_2 shown below in figure 6. H_1 is $K_4 \times K_4$, where K_4 is the complete undirected graph on 4 nodes and x is graph product. H_2 was constructed by Shrikhande [5] in connection with his work on association schemes.



H_1 $K_4 \times K_4$



H_2 SHRIKHANDE

FIGURE 6

Both of these graphs are regular of degree 6 and for every pair $\{i,j\}$ of distinct points, there are exactly 2 other points adjacent to both i and j. Thus an adjacency matrix representing either of these graphs satisfies $A^2 = 4I + 2J$ (J is the matrix of all 1's).

Assume $A^m = aA + bI + cJ$

then $A^{m+1} = aA^2 + bA + cAJ$

$= a(4I + 2J) + bA + 6cJ$

$= bA + 4aI + (2a + 6c)J$

Thus, for either graph, the multi-set of numbers derived from higher powers does not give any additional information than the original adjacency matrix. These graphs are not isomorphic (as shown below). Therefore, walk statistics, although isomorphism invariant, will not provide a canonical form.

A Russian Attempt: V. A. Skorobogatov [6].

In his paper, Skorobogatov conjectured that the following may be an algorithm for graph isomorphism.

Given graph G, node a, we construct subgraphs $G_1, G_2, \ldots$ as follows (V(G) = vertex set of graph G, E(G) = edge set of G):

$V(G_1) = a$

$V(G_{i+1}) = \{v \in V(G) \mid v \text{ adj } m \in V(G_i) \text{ and } \forall j \le i \; v \in V(G_j)\}$

$E(G_i) = \{\{x,y\} \in E(G) \mid x,y \in V(G_i)\}$

Define n_i = number of nodes in $G_i = |V(G_i)|$

u_i = number of edges in G between $V(G_i)$ and $V(G_{i+1})$

$\bar{\lambda}(a) = (n_1, n_2, \ldots, n_k, u_1, u_2, \ldots, u_{k-1})$

We can order $\overline{\lambda}^{-}(a_i)$ for $i=1,\ldots,n$ and thus get

$$\Lambda(G) = \text{ordered } \{\bar{\lambda}(a_i)\}$$

$$= \text{an isomorphism invariant matrix.}$$

Using our previous example of graphs H_1 and H_2, we see (using this algorithm) that $\bar{\lambda}(a)$ (for any node in either graph) = (1,6,9,6,13). However, closer inspection of subgraphs $G_2(H_1)$ and $G_2(H_2)$ reveals differences (see figure 7) that show H_1 and H_2 are not isomorphic. But $\Lambda(H_1) = \Lambda(H_2)$, therefore this mapping is not isomorphism indicative, and hence does not characterize graph isomorphism.

$G_2(H_1)$ $G_2(H_2)$

FIGURE 7

Given a graph $G = (V,E)$, we define the neighborhood graph $\delta_G(a)$ of a vertex $a \in V$ as

$$\delta_G(a) = (N,L) \quad \text{where}$$

$$N = \{u \in V \mid \{u,a\} \in E\}$$

$$L = \{\{u,v\} \in E \mid u,v \in N\}$$

i.e., the G_2 graph of Skorobogatov's algorithm.

Given two graphs, $G_1 = (V_1,E_1)$ and $G_2 = (V_2,E_2)$, we shall say that G_1 is <u>δ-isomorphic</u> to G_2 IFF there exists a bijection $f: V_1 \to V_2$ such that $\delta_{G_1}(a) \cong \delta_{G_2}(f(a))$ for all $a \in V_1$.

δ-isomorphic graphs are, in some sense, locally isomorphic.

Isomorphic graphs are clearly δ-isomorphic. If the converse were also true then one could construct a graph isomorphism algorithm which attempts to pair isomorphic neighborhoods, perhaps recursively. However, the example shown in figure 8 shows that δ-isomorphic graphs need not be isomorphic.

It can be seen that G is not isomorphic to H by counting 4-circuits: G has 9 whereas H has only 6.

SPANNING TREE PROBLEM

This problem is a sub-problem of the subgraph isomorphism problem. Given a tree T on n nodes and a graph G on n nodes, is T isomorphic to a spanning tree of G?

Instances of this problem range from simple (quadratic time, as in "Is the (n-1)-pointed star a spanning gree of G?") to complete ("Is the (n-1) chain a spanning tree of G?")." The proof of completeness of this latter problem is given below.

<u>Theorem</u> Hamiltonian circuit problem red linear spanning tree (Hamiltonian path) problem.

<u>Proof</u>: Given $H = (V,E)$, pick $x \in V$.

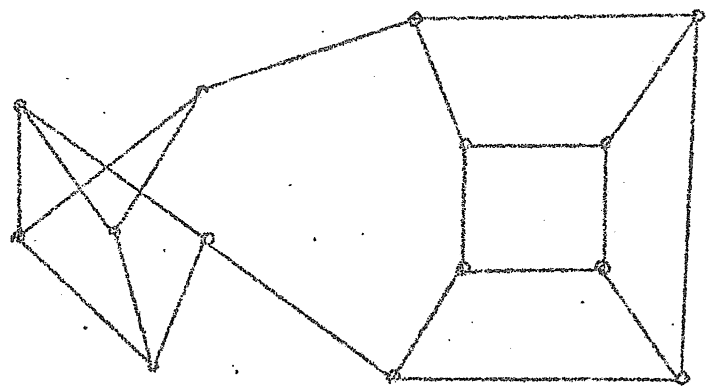
Let $W = \{v \in V \mid \{v,x\} \in E\}$.

Construct $\hat{H} = (\hat{V},\hat{E})$: $\hat{V} = V \cup \{y,\alpha,\beta\}$

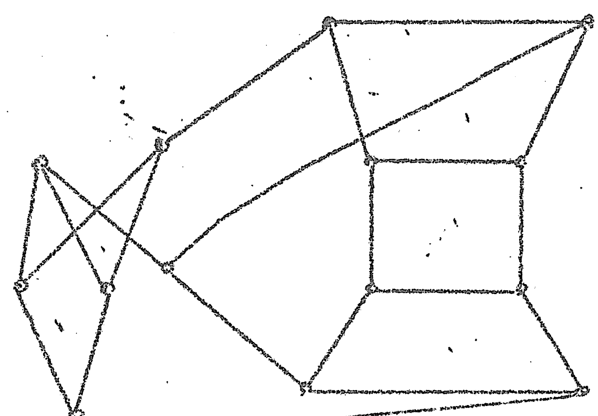
$$\hat{E} = E \cup \{\{y,w\} \mid w \in W\} \cup \{\{\alpha,x\},\{\beta,y\}\}.$$

H has a Hamiltonian circuit IFF $\hat{H}$ has a linear spanning tree.

(⇒) Assume H has a Hamiltonian circuit, say



G 9 4-circuits



H 6 4-circuits

FIGURE 8

$[x, w_a, v_1, v_2, \ldots, v_{n-3}, w_b, x]$ (the two nodes adjacent to x must be in W), then $[\alpha, x, w_a, v_1, v_2, \ldots, v_{n-3}, w_b, y, \beta]$ is a linear spanning tree of $\hat{H}$.

(<=) Assume $\hat{H}$ has a linear spanning tree, say $[\alpha, x, w_a, v_1, \ldots, v_{n-3}, w_b, y, \beta]$ (having degree 1, α and β must be endpoints of the tree), then $[x, w_a, v_1, \ldots, v_{n-3}, w_b, x]$ is a Hamiltonian circuit in H.

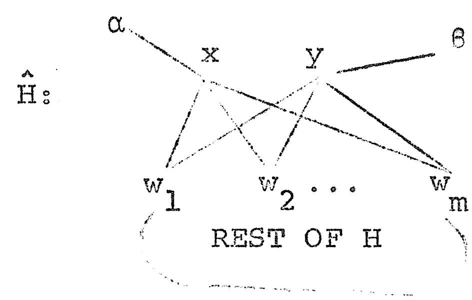


FIGURE 9

An interesting problem (for further research) presents itself: Which families of trees make this problem complete and is there a (sharp) dividing line between "polynomial" and "complete" tree families?

REFERENCES

1. Cook, S., The Complexity of Theorem-Proving Procedures, Conference Record of Third ACM Sumposium on Theory of Computing, pp. 151-158, 1970.

2. Corneil, D.G. and C.C. Gottlieb, An Efficient Algorithm for Graph Isomorphism, Journal ACM 17 1, pp. 51-64, Jan. 1970.

3. Hopcroft, J. E. and R.E. Tarjan, Isomorphism of Planar Graphs, Complexity of Computer Computations, Plenum Publishing Co., pp. 131-152, 1972.

4. Karp, R.M., Reducibility among Combinatorial Problems, Computer Science Technical Report 3, University of California, Berkeley, April 1972.

5. Shrikhande, S.S., The Uniqueness of the L2 Association Scheme, Ann. Math. Statist., 30, pp. 781-798, 1959.

6. Skorobogatov, V.A., A Decision Algorithm for Isomorphism of Nondirected Graphs, Vychislitel'nye Sistemy (Novosibirsk), No. 33, pp. 34-36, 1969.

7. Weinberg, L., Plane Representations and Codes for Planar Graphs, Proceedings of Third Annual Allerton Conference on Circuit and Systems Theory, pp. 733-744, 1965.